Weekly Report

Zhiyu Ding

September 15, 2013

1 Work scheduled in last week

The work scheduled in last week are listed as below:
1. do some tests about the new feature(volume separation), and analyze the results.
2. read the paper <Fast and Efficient Compression of Floating-Point Data>, <Fast Lossless Compression of Scientific Floating-Point Data> and <Compression Domain Volume Rendering>.
3. read the 6th paragraph of the book <Crafting Your Research Future> and the paper <How to read a paper> and form some conclusions.
4. read the course material of compression topic.
5. help Jiangang to wirte the Chinese version of compression paper.
6. read the paper <Transform Coding for Hardware-accelerated Volume Rendering>.
7. read ther paper <A Study on the Effect of Codebook and Codevector Size on image Retrieval Using Vector Quantization>.
8. reconstruct the LBG-VQ Project to be an OOP version.

2 Introduction

last week, we tested the new added feature, namely, volume separation, and analyze the results. However, the results are not as high-quality as we imagined, in terms of compression performance. So, we should keep read some papers to find out new methods to enhance the quality. At first, I reviewed some papers that I have read before to organize a whole impression of lossless(include near lossless) compression pipeline, as a basic survey. Then, two important papers are selected to be studied into the detail, they are <Transform Coding for Hardware-accelerated Volume Rendering> and <Compression Domain Volume Rendering>. Besides,I found a paper named <A Study on the Effect of Codebook and Codevector Size on image Retrieval Using Vector Quantization> maybe related to the topic as "the relationship among codebook size, codevector size and compression performance", while the result is still negative.
In order to do some comparisons and for the convenient of updating in the future as well, we plan to reconstruct the current LBG-VQ project to an OOP version, it is ongoing now.

3 Details

As we supposed,if we add a separation procedure before the existing VQ process, the result will be better. For instance, suppose we have a volume data bonsai(256X256X256), size of codebook is 256 and block size is 1X2X1, if we do VQ compression directly on the original volume, the compression ratio of VQ will be 0.12503 and the residual ratio is about 0.787. In the same setting of parameters, we gained a compression ratio valued 0.12524 and a residual ratio valued 0.945 when we deal with a 128X128X128 volume data, such as hydrogenAtom. 256 size codebook will acquire better result when dealing with relatively small volume datasets rather than facing large scale ones. Based on this phenomenon, we assume that to separate

large scale volume into smaller ones first, then to do the VQ compression successively will achieve a better compression performance. Here, we separate the 256X256X256 size volume into 8 128X128X128 ones. For each of sub-volumes, a 256 size VQ compression is executed. As the assume, we should get a very good result, such as the result of hydrogenAtom mentioned above, for each sub-volume. Unfortunately, the result is not on its' supposed style. We found some results of sub-volumes are good while others are bad and the final average results are nearly the same when comparing with counterparts of the direct method. It is the different data distribution and characteristic that the results dependent to the corresponding sub-volume. We will try a new solution after the OOP reconstruction of VQ project, to separate large scale volume using a uniform blocksize (eg.2X2X2) first, eg. separate a 256X256X256 size volume into 8 128X128X128 volumes, followed by the VQ compression. In the third step, we select the low-quality sub-volumes, to assign a refined block size(eg.1X1X1)to these sub-volumes. Finally, recompute VQ compression for these sub-volumes. We will also study the relationship between the data distribution as well as some other characteristics. If we will get this kind of relation, we can do a pre-compute step before VQ compression, then do the separation and compression.

The current lossless compression methods for volume data mainly focus on coding and transforming approaches, such as Wavelet-Transform, huffman Encoding, Run Length Encoding and Lempel-Ziv-Welch. However, these kind of methods always trapped in real time decompression in GPU because of a very high number of texture accesses. As a construct, Vector Quantization based methods achieve very fast decoding speed, because all the operations in GPU are only two texture fetching usually. This kind of methods are totally considered as an asymmetric compress method, namely, slow speed coding(offline, preprocess) and simple, fast decoding(on-the-fly, real time). However, VQ based methods have there own drawbacks. In general, Vector Quantization based methods are defined as a lossy compression method. To the best of my knowledge, all the work that VQ-based are in the lossy compression field. Hence, the main and core contributions of our approach(if goes well) should be simply concluded as:

1. The first VQ based near lossless compression approach for floating-point volume data.
2. A relativly high compression ration. Using VQ based method will obtain a high compression ration, but compression quality may vary according to the distribution of data. So, to create an new VQ based approach that can achieve a relatively high-quality compression result(near lossless) while still keep a high compression ratio in lossless compression field for floating-point volume dataset is a meaningful work. Further more, if we can obtain the rules behind the data distribution and compression quality, even for specified kind of volume data, I think it will be a respectable job. The last point we should keep our eyes on is that several papers, including <Transform Coding for Hardware-accelerated Volume Rendering> have mentioned that 4X4X4 is a good choice for block size. This conclusion is obtained from large experiments rather than some solid Mathematical regulations.

4 Conclusion and Future Work

As a conclusion, we implemented the new feature, did the correctness tests and analyzed the results. The results are not as good as we assumed;thereby leading to the new scheduled solution. After a systemic review and study about dozens of compression topic related papers, a general concept contains all the major branches is formed as well as the main contributions of our work(if goes well). In the future, I will:

1. reconstrruct an OOP version LBG-VQ project, if done, to implement the new solution and test it.
2. read papers and course materials to search new directions and ideas.
3. VIS 2013 paper collection will be finished in one day.
4. ask Jiangang to implment the HVQ method(<Compression Domain Volume Rendering>).
5. if needed, to implement the TVQ method(<Transform Coding for Hardware-accelerated Volume Rendering>).
6. study the relationship between compression quality and data distribution by using histogram first.

References